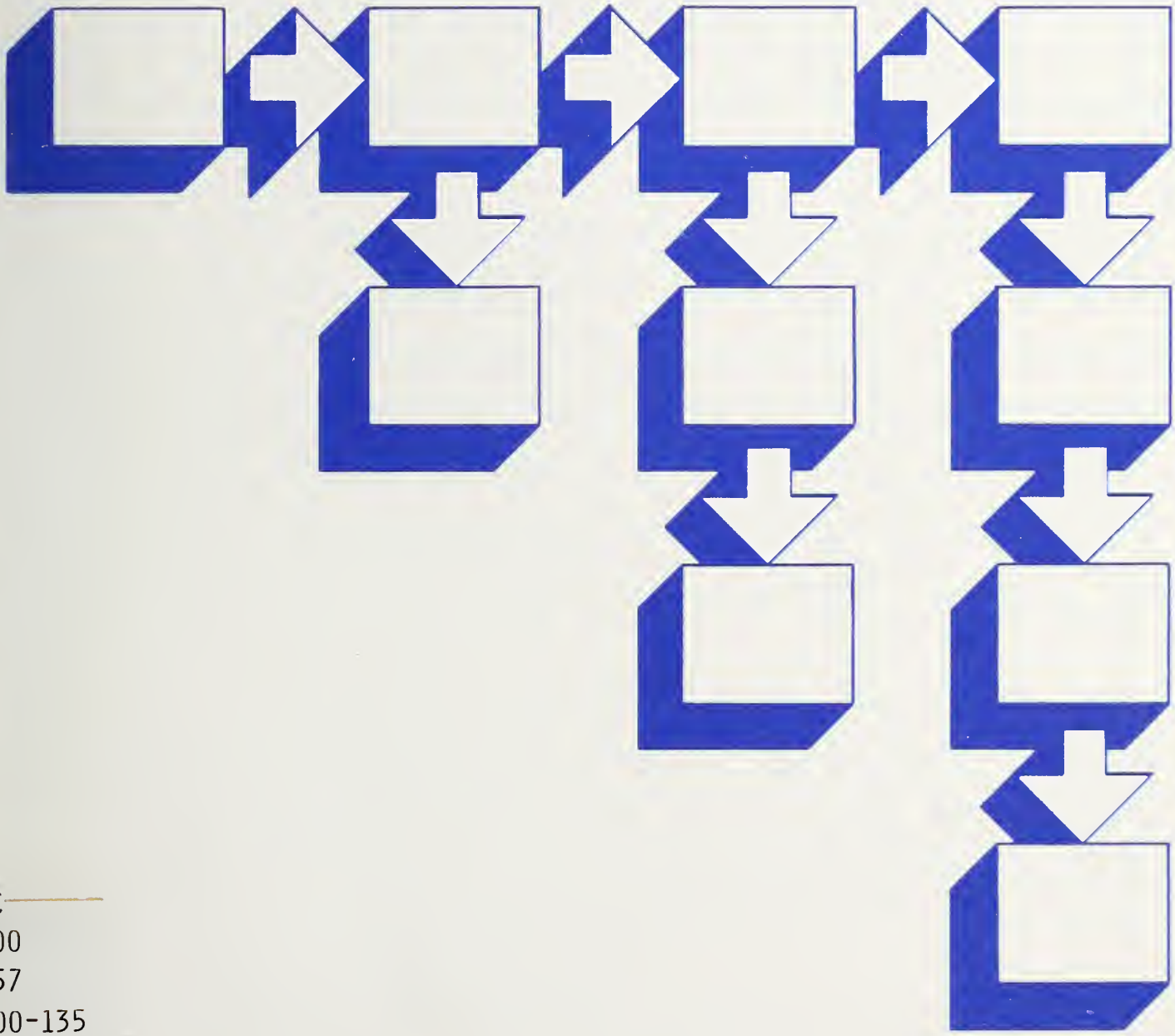
Computer Science and Technology

NBS Special Publication 500-135

Integrated Software for Microcomputer Systems



The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the Institute for Computer Sciences and Technology, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[2]

The Institute for Computer Sciences and Technology

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

- Programming Science and Technology
- Computer Systems Engineering

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-country scientific themes such as nondestructive evaluation and phase diagram development; oversees Bureau-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following Divisions:

- Ceramics
- Fracture and Deformation [3]
- Polymers
- Metallurgy
- Reactor Radiation

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2]Some divisions within the center are located at Boulder, CO 80303.
[3]Located at Boulder, CO, with some elements at Gaithersburg, MD.

Computer Science and Technology

NBS Special Publication 500-135

Integrated Software for Microcomputer Systems

Lynne S. Rosenthal

Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Bureau of Standards
Ernest Ambler, Director

Issued January 1986

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

Library of Congress Catalog Card Number: 86-600500
National Bureau of Standards Special Publication 500-135
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-135, 38 pages (Jan. 1986)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1986

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

ABSTRACT

It is not unusual to find at least one computer system in every office today. These systems are being used by technical, managerial, and administrative personnel to assist them in the performance of their jobs. Various types of *office tools* have evolved to facilitate information processing, and, consequently, improve the productivity of these office workers. This report concentrates on one type of office system and office tool: the microcomputer and integrated software products.

Integrated software products combine several applications within a single program and enable information to be shared between the applications. This report defines five approaches to integration:

1. the multifunction application approach,
2. the product suite approach,
3. the software integrator approach,
4. the operating environment approach, and
5. the background utility approach.

Each of these approaches is designed to achieve different objectives by emphasizing the power and importance of the features of each approach. Consequently, there is no best approach to software integration. The selection of an approach depends on the application requirements, current system configurations, and personal preferences.

Selecting an integrated product begins by considering the various approaches to integration and determining which one is most appropriate. Subsequently, the products within the chosen approach are evaluated against a preestablished set of criteria relating to the product design, technical capabilities, and product quality. Careful selection of an integrated product will insure that the benefits to be gained from its use can be achieved.

Contents

List of Tables

1 INTRODUCTION

No longer does office automation simply mean word processors for secretarial and clerical personnel. Today, it implies the use of microcomputers, mini- and mainframe computers, and other technologies to assist office workers (administrative, managerial, technical and clerical personnel) perform their jobs. Additionally, a host of *office tools* have evolved to facilitate the interchange of information in this automated office environment. These tools are directed at improving the productivity and effectiveness of the office worker and improving the timeliness and accuracy of the information on which they depend.

The tools of an office automation system must be able to work together and integrate different types of information from heterogeneous sources. For example, the tasks performed to produce a report or make a decision may involve:

1. accessing information in different forms, such as, text, graphics, images, or voice,
2. using an office tool to process information,
3. combining one tool with another (passing information from one tool to another for further processing), and
4. transferring the information from one computer system to another and using the tools of the latter system.

One method for accomplishing these tasks is by providing a flexible environment, in which an individual can select and combine the office tools required to handle the information that needs to be processed. This method is called integration. There are two basic types of integration: vertical integration or relaying information between computer systems (e.g., between a microcomputer and mainframe computer) and horizontal integration or linking functional applications together (e.g., combining a data base manager with a spreadsheet).

This report concentrates on horizontal integration and in particular, integrated software products for microcomputer systems. These products combine several office tools or applications within a single program. The report defines and describes five ways of achieving software integration, explores the reasons for using integrated software products, and concludes with selection criteria.

1.1 Background

If jobs consisted of isolated, individual tasks, there would be no need for integrated software. Single application software programs would be sufficient to meet the computing needs of the user. However, many jobs involve multiple, heterogeneous tasks while processing the same set of data. Additionally, as user expertise and sophistication grows, the tendency to expand the scope and complexity of a job increases. Users want to combine

tasks and perform operations that the software was not originally designed to do. They seek software that will provide multiple task applications and eliminate the inability or difficulty of sharing data between application programs and the need to learn command languages of several programs. Several forms of integrated software are emerging as viable solutions to software application problems.

Integrated software offers the user an alternative to working with one task and one data file at a time. These software products can split the screen into multiple subscreens called windows, enable data in several files to be viewed at once, combine applications, and can accomplish this using a common command structure. Providers of integrated software also claim that such software increases the productivity of the user by being faster and easier to use.

1.2 Historical Perspective

Users first began to use personal computers with standalone software packages that performed a single task. These packages helped to change the way information was processed by automating tasks that were previously not automated. Since jobs are not defined as single tasks, but rather incorporate several tasks, it was not long before users wanted to combine data and applications. For example, while preparing a budget report, an analyst wants to combine information in a personnel data base with a budget-related spreadsheet, create a graphical representation of the spreadsheet, and put the graph and parts of the spreadsheet into a report. An initial level of integration was achieved through file compatibility, the ability of one application to read and manipulate the files of another. The output of one application package (e.g., word processor) becomes an intermediate data file to be used as the input file to another application package (e.g., spelling checker). Typically, the intermediate file is written to disk prior to use as the input file, and may require conversion to another data format (code representation and layout). File-level integration is not always possible and is often complicated and time-consuming to perform.

First generation integrated software products evolved as an efficient way to eliminate intermediate file creation and to combine data and applications within a single program. Users could switch from one application to another and use the same data without exiting the software product. However, these products typically were based on a specific application (i.e., spreadsheet) and offered a limited range of functions for the companion applications. As these integrated software products gained popularity, many users desired additional applications and a fuller range of capabilitites. Particularly desired was a communications capability. During this time, technology advances in the forms of cheaper microprocessors, system memory (random access memory, RAM), and internal hard disks, along with the increasing popularity of user interfaces such as, pull-down menus, windows, mice, and icons, gave rise to the development of second generation integrated software. Taking advantage of these advances, second generation integrated software provides greater flexibility, power, and functionality than that of the first generation. Moreover,

new methods for achieving integration developed and are included in second generation integration.

1.3 Document Structure

This document is divided into five sections and an appendix. Chapter one provides background information. Chapter two describes the meaning of integration and defines five ways integration can be achieved. Selection criteria for choosing an integration approach and product is presented in chapter three. Chapter four concludes with an overview of the marketplace and some future directions. References are contained in section five and the appendix lists currently available integrated products.

1.4 Disclaimer

Because of the nature of this report, it is necessray to mention vendors and commercial products. The presence or absence of a particular trade name product does not imply criticism or endorsement by the National Bureau of Standards, nor does it imply that the products identified are necessarily the best available.

2 WHAT IS INTEGRATED SOFTWARE

Integrated software means different things to different software vendors and users. It has become an umbrella term to describe software that permits data, applications, and/or programs to be combined and manipulated within a single environment (e.g.,program). The major differences between products stem from what is actually integrated and the method for accomplishing the integration. Variations on the word integration have been used to distinguish between the different approaches to integration, e.g., integrated software, integrating software, software integrators. For this report, the term integrated software is used to refer to all these types of software products, in other words, the entire class of software integration products. The distinction between the different approaches to integration is discussed in a later section. References to specific types of software and integration approaches will be used whenever appropriate.

2.1 Attributes of Integration

Not all software products that claim to be integrated, are integrated. Many of the features associated with integration are not a necessary condition for integration. For example, menus, windows, mice, and macros are incorporated in many integrated products, but are also present in non-integrated products. To understand what integrated software is or is not, it is helpful to examine the attributes that need to be present to make the product integrated. Based on our research and for the purposes of this report, in order for a software product to qualify as *integrated*, it must exhibit two attributes:

- Multiple applications - the union of two or more applications within an environment. The transition from one application to another is accomplished without exiting the controlling environment. Typically, three to six applications are combined, with word processing, spreadsheet, and file management as the predominant applications.

- Data sharing - the ability to move information from one application to another. The transfer is performed without the user having to retype the data, convert the application's native data format to another format, or use extraneous file conversion software. The environment controls and performs the movement of data between the applications. Typically, the shared data is either copied from the source to the destination or is centrally located and accessible by any application.

The following attribute is not a necessary condition for integration, however; it is often provided by integrated products.

- Common user interface - the use of similar command structures and syntax throughout the software product. A command, function key, or typewriter key will perform the same task regardless of the application module in which it is executed. This consistent presentation and functionality of the command words, menus, and/or screen

designs facilitate the learning and usage of additional applications once one has been mastered.

2.2 Approaches to Integration

There are several approaches to integrated software. Each method is designed to achieve different objectives and stress the power and importance of different features. Consequently, each has its strengths and weaknesses and is appropriate for a variety of users and applications. No absolute classification scheme exists. Therefore, this report defines a categorization scheme for the purpose of better understanding the differences, and at times nuances, of integrated product features. The following five sections describe these categories:

1. multifunction application,
2. product suite,
3. software integrator,
4. operating environment, and
5. background utility.

To demonstrate the integrated software attributes (mulitple applications, data sharing, and common user interface) present in the categories of approaches, Table 1 provides a matrix of approaches and attributes.

The section organization used to describe each category is designed to demonstrate the similarities and differences of the categories. The organization is as follows: definition of the category, examples of products included in the category, description of key advantages, and description of key disadvantages.

INTEGRATION APPROACH	ATTRIBUTES		
	Multiple Applications	**Data Sharing**	**Common User Interface**
MULTIFUNCTION APPLICATION	√	√	√
PRODUCT SUITE degenerate case (1 application)	√	√ √[1]	√
SOFTWARE INTEGRATOR degenerate case (1 application)	√	√ √[1]	
OPERATING ENVIRONMENT degenerate case (1 application)	√	√ √[1]	√[2]
BACKGROUND UTILITY	√	√	√

[1] Data can be shared between multiple copies of data from a single application.

[2] A common user interface may not apply to all applications within this approach.

Table 1: Integration approaches and their attributes

2.2.1 Multifunction application

The multifunction application or all-in-one application is a single software product that consists of several different applications. The applications may vary, but usually include: spreadsheet, data base mangement or file management, graphics, and word processing. The applications are predetermined and preassembled by the vendor, thus the user has no choice in selecting the specific applications contained in the product. Products in the multifunction application category can be thought of as a *bundled* product.

Examples: 1-2-3, Framework, Enable, Jazz

Advantages

- Data can easily and quickly be copied between applications. A referencing capability enables copied data to be automatically updated if the original data is changed.

- Switching from one application to another is accomplished quickly and with little effort.

- The screen can be divided into subscreens called windows, with each window independently displaying data from the same or different applications. When you switch from one window to another, the activity in the window is suspended until you return to it. This allows the user to closely coordinate different but related tasks and facilitates data transfer between applications.

- The consistency of the command structure and documentation enable the user to immediately use new or infrequently used applications once one application is learned.

Disadvantages

- A large amount of system memory (RAM) is required to run these programs. To process a large database, spreadsheet, or document, additional RAM beyond the minimum specified by the product is necessary. Usually, 512K RAM is recommended for good performance.

- Only one application within an multifunction application product is likely to be as good as its standalone product equivalent. The other applications are limited to basic functions and lack many of the sophisticated features found in the standalone product.

- Often these products are very complex and require a great deal of time and effort to learn their full range of capabilitites, causing users to become impatient, discouraged, frustrated, or intimidated.

2.2.2 Product suite

The product suite or program family consists of a series of separate programs designed to work together and share data. The products, generally from one vendor, share a common command structure, data format, and/or data base. Since applications for the product suite can be purchased as a set or individually, the suite can be considered an unbundled product. Although it is possible to have a product suite with only one application, two attributes of integrated software, multiple applications (a necessary attribute for qualification as integrated) and common user interface, are not present in this degenerate case. Thus, the degenerate case does not qualify as integrated.

Examples: Smart Software, Perfect Series

Advantages

- The specific applications included in the product suite are chosen by the user. The user tailors the suite to his current needs, adding new applications only as they are needed. This building block approach eliminates the extra RAM and money spent on applications not needed or never used.

- Each application within the product suite offers a full range of features and depth of functions. The quality of the application is equivalent to its standalone product competitors.

- All applications use the same basic commands, manuals follow the same basic pattern, and installation of each component is consistent. This consistency makes the product suite easy to learn.

Disadvantages

- The process of transferring data may be complicated, requiring the user to perform a complex sequence of commands. While intuitive, it must be pointed out that data sharing is guaranteed only for those applications developed and contained within the product suite.

- Although the user can select the type of applications contained in the product suite, he is restricted to software written specifically for the suite. The user may need an application that is not available for the suite, or he may have to settle for an application package that does not contain all the features desired.

- Only one application can occupy the screen at a time. This prevents a user from creating windows that display the data of different applications, e.g., concurrently displaying a word processing document and a spreadsheet. However, it is possible to create windows that display multiple copies of data from the same application.

2.2.3 Software integrator

The software integrator or applications integrator creates an *open environment* within which user-chosen application software packages can operate and exchange data. The participating packages, usually from different vendors, retain their identity and capabilities. The software integrator supervises the operation of the application software and loosely ties these independent applications together. A software integrator that contains only one application package is considered a degenerate case, and as in the case of the product suite loses attributes required to be defined as integrated software.

Examples: DesQ, Windows

Advantages

- The applications can be selected from off-the-shelf standalone packages, custom or special purpose applications, and even other integrated packages. This facility enables users to continue to use applications software already in their software inventory and to add new applications that match their needs.

- Applications with incompatible data formats can exchange data. This expands the functionality of the application by supplementing its functions with those included in other applications. Data produced by one application can undergo further processing by another application. For example, if a spreadsheet package lacks a word processing capability, then through the system integrator, a data file created by the spreadsheet package can be transferred into a specific text processing package and edited.

- Windows can independently display data from the same or different applications. The activity in the window stops when a second window is activated and restarts when control is returned to that window. This allows the user to closely coordinate different, but related, tasks, and facilitates data transfer between applications.

Disadvantages

- There are limitations on the data that can be exchanged between applications. Data is not transferred in context, that is, the positions and relationships of numbers and text are not maintained. For example, spreadsheet formulas and row and column positions are not transferable. Thus, to maintain the layout and intent of the data, it may be necessary to retype or reformat transferred data.

- As with the multifunction application category, a large amount of RAM is required to run these programs. Additionally, a megabyte or more of hard disk storage may be necessary to store the system and application files.

- Several different applications are combined by the system integrator, each with its own set of commands, screens, and documentation. The user is forced to learn the

commands and procedures of every application. This can become confusing and frustrating especially when the meaning of commands differs from application to application. For example, in a given word processing application, data base application, and spreadsheet application, the function key, F5 *replaces* a section of data, *erases* a section of data, or *goes to* another location, respectively.

- Not all applications will work properly in the system integrator environment. Programs that are not *well-behaved*, that is, bypassing normal methods for reading and writing to the screen, may not work correctly or completely. For example, an application program which in order to speed up operations, issues commands directly to the computer, circumventing the operating system's routines, would not be well-behaved.

- Prior to using these products, a file that defines the applications, what they do, and how they do it is required. To develop this file, a user must have considerable knowledge of each application, its memory requirements, cursor control, and other operational specifics. Most users do not have this level of expertise and will not be able to create the system integrator's set-up file. To address this disadvantage, the set-up files for some of the *popular* applications have been included in system integrator packages, thus eliminating this problem.

2.2.4 Operating environment

The operating environment is either an operating system or a software program that submerges the actual operating system. It controls and coordinates the applications and their access to peripherals (e.g., video display, memory, disk drives). In contrast with the *open environment* of the software integrator, only applications designed specifically for the operating environment can run and exchange data. Interfaces into the environment (hardware and/or software) are provided to enable third party vendors and users to develop additional applications. An operating environment with one application is considered a degenerate case, and as such loses attributes required to be defined as integrated software.

Examples: Gem, Topview, Macintosh

Advantages

- The user can add applications to the environment as they are needed: thus, providing flexibility and eliminating unused or unwanted applications.

- Data is easily shared between applications. Data is transferred in context, i.e., the positions and relationships of numbers and text is maintained.

- Windows can independently display data from the same or different applications. The activity in the window stops when a second window is activated and restarts when control is returned to that window. This allows the user to closely coordinate different, but related, tasks and facilitates the data transfer between applications.

- A unified command structure accelerates the learning of the applications within the environment.

Disadvantages

- As with the multifunction application and software integrator a large amount of RAM is required to run these programs. A hard disk with at least one megabyte of storage may also be required.

- The user is restricted to application software written specifically for the environment. Therefore, a user-required application may not be available for the environment or an application provided may not contain all the user-desired features.

2.2.5 Background utility

The background utility or desktop manager is a multifunction program designed to be used in conjunction with applications software, supplementing them with functions such as a notepad, calculator, and calendar. The utility remains hidden in the computer's RAM until needed, while another program runs in the foreground. The utility can be invoked at anytime, temporarily suspending operation of the current activity.

Understanding a background utility can best be illustrated by an example.

> While typing a table with the word processor, the user needs to add a list of numbers; or while using a spreadsheet, the phone rings and the user wants to write a short note or check his calendar. Ordinarily, this means interrupting the application, to use a hand calculator or write a note. With the background utility the user quickly switches to an on-screen calculator, notepad, or electronic appointment calendar and then returns to the application at the point of interruption.

Examples: Sidekick, Desk Organizer, Spotlight

Advantages

- The electronic substitutes for desktop items are always available to the user, even while in the midst of another application. The application can be easily interrupted and remains intact while the utility is used. This eliminates the time-consuming chores of exiting the application, saving the work previously accomplished, and reloading the application and data when ready to resume.

- Most people do not need the power and features of a fully loaded integrated product. Aside from using one major application (e.g., word processing, spreadsheet, communications), a user may want the ability to do other limited functions (calculating, editing, filing, etc.). The desktop utilities offer these basic functions, integrating them with the major applications and better yet, are relatively inexpensive to purchase and require a modest amount of RAM.

- Data can be exchanged within the utility as well as between the utility and an application. Thus, data created in either the application or utility can be copied and processed by the other.

Disadvantages

- With the utility resident in RAM, the quantity of RAM left to load and run other software packages is reduced, and may not be sufficient.
- Some software packages do not run if another program is also resident in RAM. Since the utility is always RAM resident, the user can not use both the utility and this type of software package.
- Unless installed on a hard disk, the utility program disk must remain within the system to operate the program. The computer's disk drive is preempted and unavailable to another disk. This becomes a problem for users who maintain separate program and data disks, each of which occupy a disk drive.

3 CHOOSING INTEGRATION

Integrated software has captured the interest of personal computer users with its innovative features, powerful capabilities, and the potential for new uses. A demonstration of any one of these packages is impressive. And, if we believe the advertising claims, integrated software is the answer to all of our software needs. No wonder everyone wants one. But, is the need real, and if so, which category and product within that category is the best?

The guidelines presented in the following sections are intended to assist the reader in determining the answers to these two questions. Subsequent to the discussion on whether to use integrated software, criteria to consider when selecting an integration approach and finally, specific integrated products are presented. The guidance is intentionally general in nature, since the answers are dependent upon many factors, including the type of application and user. It is not a checklist or a cookbook approach to evaluating integrated software, but a discussion of key issues that should be addressed. This is not intended to be a comprehensive list of all possible issues but rather a starting point. It should be noted that a complete software selection methodology is not included. The reader is referred to Section 5 for additional information on selection.

3.1 Is Integration Needed?

Integrated software is not appropriate for every situation. Before making the decision to buy an integrated product, carefully examine its costs, what it can and can not do, and how it can alter the way you work. As with the selection of any software application product, at least a minimal requirements analysis should be performed. Considerations should include: a determination of application requirements, a review of current and/or required hardware and software, and an evaluation of candidate software packages and alternatives. Compare the benefits to be gained from using an integrated product versus a standalone application product. Keep in mind that the benefits of integration are not achieved overnight. Integration may make work easier and improve the quality of the work, but will take time to master and apply.

The decision to use an integrated product will be influenced by current microcomputer systems and applications software usage. It may be necessary to replace or augment existing hardware in order to meet the configuration requirements of an integrated product. Additionally, use of an integrated product may supersede, complement or incorporate existing application software. Any changeover to a new system and software is likely to be disruptive, consuming time to install, to learn, and to convert existing files. However, the concerns raised in this paragraph would also be relevant for any standalone product added to an existing microcomputer system.

There are a number of potential benefits that may be associated with the use of integrated products. It must be stressed that the benefits are only potential due to a number

of associated trade-offs between using integrated products and individual software products. There is no quantitative data or study results to substantiate the debate around the tradeoffs discussed below. However, their existence must be recognized and factored into the decision to use integrated products.

There are six classes of potential benefits to using integrated products: versatility of switching between applications, data sharing, ease of use, reduced user resistance to new software, reduced organizational support, and cost.

3.1.1 Versatility

The strength of integration lies in its versatility. The power of an integrated product's individual applications is augmented by the immediate availability of its companion applications. Users who switch between tasks or need additional information from another application to complete a task should consider using integration. It allows the user to work in a natural way, imposing no penalties for changing the order in which work is done. Interruptions can be dealt with as they arise by temporarily putting aside the current application and switching to another application or data file. The integrated product provides an effective way to handle this situation, minimizing the time and keystrokes involved and retaining the status of the original application and data. Although this flexible environment is desirable, too much switching can be unproductive; the interruption and creation of too many tasks may impede the completion of older ones. Another concern is that a user's eagerness to integrate will result in application requirements being sacrificed; the applications contained in the chosen integrated product should be adequate to meet the requirements of the job.

Using integration should provide the potential to increase personal productivity by helping users do their job. The right mix of integrated applications should assist the user in solving problems and performing the analysis needed to make decisions quickly and easily. The integrated components should be useful and not get in the way or idly occupy RAM or disk space. If the extra functionality is more than is necessary, it may be better to use either an integrated product with fewer functions or a standalone product.

If the computer is used primarily for one task or several unrelated tasks, there may be little incentive to switch to an integrated product. A standalone product may be more suited to the single task than the corresponding application of an integrated product. Standalone products may provide a range and depth of features not available in an integrated product. Even if one of the applications within an integrated product contains the required features to do the task, the additional overhead from the multiple applications, the command language, and the memory requirements may not be worth migration to an integrated product.

3.1.2 Data Sharing

One of the most prevalent reasons to choose integrated software is data sharing, shar-

ing information between applications. The ability to share data promotes the recency and accuracy of data. Before deciding that this capability makes an integrated approach essential, examine the actual need for this capability. Consider the frequency in which data will be transferred between applications and the difficulty in sharing data between unintegrated software products. If data will not often be shared and if provisions for importing and exporting data between packages are provided, an integrated approach may not be warranted. Surprisingly, the methods for importing and exporting data are sometimes easier to perform than the data sharing methods furnished in some of the integrated products. On the other hand, many integrated products offer the advantage of data referencing, the automatic update of copied data to reflect changes made in the orginal data.

The location and organization of the information to be shared should be considered. Integration may be of limited use if information is scattered between different computers, e.g., mainframe data bases, dedicated word processors, and microcomputer spreadsheets. Aside from the ability to communicate with another computer and download files, only a small number of integrated products can currently handle and process information located on a different computer.

3.1.3 Ease of Use

Integrated products often offer a consistent philosophy for combining and presenting applications, e.g., all applications look and work the same way. This consistency can reduce the learning time per application. A user can learn one application and apply this knowledge to the other applications in the product. In reality, however, integrated products may not be easier to learn initially than standalone products. Integrated products are often complex, providing capabilities and features that may be daunting to a user. Users may not reach the levels of expertise required to maximize the benefits to be gained from using an integrated product. Consequently, users may find the specificity of standalone products (i.e., one application per product) less intimidating and easier to learn.

3.1.4 Reduced User Resistance

Whenever a new product is introduced into the office, it is not unusual to experience user resistance. Some users may resist surrendering their current software for the integrated products. They may feel that the integrated products lack the specialized features and capabilities needed to accomplish their job. On the other hand, the use of integrated software has the capability of reducing the amount of user resistance or at least the number of occurrences of resistance. For example, if an integrated product combines three applications, then three standalone software products may be required to provide those same applications. When the integrated product is introduced, there is a single opportunity for user resistance. However with individual products, user resistance is likely to occur every time a new product is introduced, in this case, three separate times.

3.1.5 Reduced Organizational Support

Many organizations are finding it easier to support integrated products than standalone products. One integrated product can provide a variety of applications that will meet the needs of a diverse group of users. Training and support can be generalized and applicable to all users, as opposed to supporting multiple software products, where each product addresses a different application and audience.

3.1.6 Cost

In general, the cost per application of an integrated product is lower than the unit cost of the equivalent number of standalone products. However, cost savings should not be dwelt upon since the price of software, both integrated and standalone, varies between vendors and can range from no cost to hundreds of dollars.

3.2 Selecting an Integration Approach

Once the decision to use integration is made, what type of integration and which product should be chosen? As expected, each vendor has his own approach to integration and each user has his own requirements and expectations. Consequently, there is no best integrated product or method of integrating software. The choice depends on the application requirements, current system configurations, and personal preferences.

Begin the selection process by considering the various forms integration can take and determine which one is most appropriate. It is not necessary to limit your selection to the use of a single integration approach; instead, explore the possibility of using a combination of approaches to solve your integration needs. For example, if a calendar and notepad capability is desired while using a product suite lacking these capabilities, combine a background utility with the product suite. Another possibility might entail operating an multifunction application package under a software integrator, so that the capabilities of another software product might be combined with or substitute for an application within the multifunction application package.

The remainder of this section consists of statements and questions aimed at assisting the reader in choosing an appropriate type of integration. Emphasis is on selecting an integration approach and not on choosing a specific integrated product. Criteria oriented toward selecting an integrated product are discussed in the next section. The statements and questions are classified into broad categories: application, data exchange, user interface, configuration, and cost. Following each set of statements is a chart consisting of related questions and the integration approach(es) that answer the question. Try to associate a level of importance or a desirability indicator to each question. A $\surd$ in the approach's column indicates that it answers the question affirmatively. Due to the variability of individual products within a category, it is possible for one category to have $\surd$'s in seemingly contradictory statements. The integrated products within an integration category with a

large number of affirmative or desirable responses which meet user requirements should be given serious consideration.

3.2.1 Application

Each integrated product can consist of a different set of applications. The applications can be bundled, available only as a set, or unbundled, individually available. Determine who selects the applications that will be integrated and how well the chosen applications will work together as an integrated product. The quality of applications can differ among applications of the same functional type as well as between different applications of the same integrated product. Insure that the features and capabilities for the applications are adequate for the job.

	Multifunction Application	Product Suite	Software Integrator	Operating Environment	Background Utility
1. Is the selection of applications –					
a. predetermined and preassembled by the vendor,	√				√
b. user chosen and assembled,		√	√	√	
c. user chosen, but limited to applications that meet specific conditions, or		√		√	
d. user chosen and not limited by specific conditions?			√		
2. Is the range of features and depth of capabilities –					
a. limited in some of the applications, or	√				√
b. as good as standalone product equivalents for all the applications?		√	√	√	
3. Synergy of applications: Do the applications –					
a. all work correctly and completely, or	√	√			√
b. work properly only if "well-behaved"?			√	√	

3.2.2 Data Exchange

Integrated software should provide a means of moving data between applications of the same integrated product. The degree of difficulty and user involvement required to perform the transfer and the relationship between the source data and the copy can differ among the integrated products. Examine both the process of transferring data and the characteristics attributed to the transferred data.

	Multifunction Application	Product Suite	Software Integrator	Operating Environment	Background Utility
1. After the data to be transferred is identified, is the actual transfer –					
a. performed automatically, with no user involvement, or	√			√	√
b. procedural, requiring the user to perform a series of operations?		√	√		
2. Data path: Is the data transferred –					
a. directly between the two applications, or	√				√
b. passed through an administrator or manager program which controls the transfer?		√	√	√	
3. Referencing capability: If changes are made to the original data, is the copied data –					
a. updated automatically to reflect the changes, or	√			√	
b. unchanged, requiring changes to be manually replicated?		√	√		√

3.2.3 User Interface

Integrated products contain features and capabilities that make them easier to learn, to use, and to remember how to use. Although the features and capabilities included in the integrated products can vary, products with the same approach to integration will have several features and capabilities in common. For each type of integration, examine command and documentation consistency, screen presentation, and application concurrency - the ability to switch between applications without exiting either application. Insure that the features and capabilities will create a workable environment, enabling the user to work in the most effective way.

	Multifunction Application	Product Suite	Software Integrator	Operating Environment	Background Utility
1. Are the command formats and documentation –					
a. consistently presented and functionally the same for all applications,or	√	√		√	√
b. different for each application?		√			
2. Is the switch from one application to another –					
a. a direct transfer of control, (usually fast), or	√				√
b. indirect, with a manager program controlling the transfer (usually slow)?		√	√	√	
3. Windows: Can the screen contain multiple windows with –					
a. different types of applications in each window, or	√		√	√	√
b. only the same applications in each window?		√			

3.2.4 Configuration

The hardware (e.g., memory, graphic board, monitor, etc.) or software (e.g., operating system) required to operate an integrated product varies from product to product, irrespective of the type of integration. Although some types of integration require more RAM than others, more precise statements regarding the hardware or software requirements are difficult to make. Although important in the determination of an integration approach, emphasis on these requirements should be deferred until the identification of specific integrated products.

Prior to using an integrated product, the product will need to be installed. This process can be as easy as inserting a disk into the disk drive or as involved as writing an information file that defines the hardware and software environments. Determine the level of skill required to install and set up the integrated product.

	Multifunction Application	Product Suite	Software Integrator	Operating Environment	Background Utility
1. RAM requirements: Generally do the products require –					
a. large amounts of RAM to run, or	√				
b. less amounts of RAM (depending upon the number of applications in the product?		√	√	√	√
2. Installation of the products –					
a. requires a person with knowledge of the applications' operating specifics to write the product's configuration file, or			√		
b. requires little knowledge beyond understanding the basic procedures for operating the computer system?	√	√		√	√

3.2.5 Cost

Because each type of integration offers different benefits, because each vendor has his own pricing algorithms, and because prices between products are often negligible, price should not be a major factor in choosing the type of integration. However, the price of individual applications and custom programmed applications will generally cost more than a bundled set of applications. Examine how the applications are selected and provided within each type of integration.

	Multifunction Application	Product Suite	Software Integrator	Operating Environment	Background Utility
1. Is the cost – a. fixed price for an entire product (including all applications), or	√				√
b. based on a separate price for the base product with additional costs for each added application?		√	√	√	

3.3 Selecting Integrated Products

A large number of integrated products are available in today's marketplace. Given this plethora of choice, the user must develop criteria to evaluate the products and to choose the *best* product available for a particular purpose or at least eliminate those products which are unacceptable.

This section describes criteria for evaluating integrated products; these criteria are grouped based on their relationship to:

- the product design, those technical attributes that reflect the underlying structure of the product,
- the product features, those technical attributes that reflect the capabilities of the product, and
- the quality of the product, the non-technical aspects of the product.

Table 2 lists the criteria by these criteria types. As with previous sections, this list is not exhaustive, but contains many of the more salient concerns with respect to selecting integrated software. The reader is encouraged to add to the list and to establish a level of importance for each criteria. These importance levels or desirability indicators, usually based on application requirements and user preferences are helpful in evaluating and ranking the different products.

CRITERIA TYPE		
Product Design	**Product Features**	**Quality**
Memory Usage	Application Breadth	Learning Aids
Data Exchange	Programmability	Product Quality
Storage Allocation	Multitasking	Vendor Support
Product Bias	Windows	
	Hardware Required or Supported	

Table 2: Listing of criteria by criteria type

3.3.1 Product Design

Product design includes technical attributes that reflect the underlying structure of the product. The attributes include: memory usage, data exchange, storage allocation, and product bias.

Memory Usage

An integrated product can be either memory-oriented or disk-oriented. A memory-oriented product limits the size of data to the amount of available RAM; the entire data file must be loaded into memory in order to be processed. There is no provision to load a portion of the data and retrieve additional portions from the disk as needed. It naturally follows, that as the amount of RAM increases, so does the size of the file that can be processed. In contrast, disk-oriented products do not limit the size of the data file. If a data file exceeds the RAM size, a portion of that file is loaded into RAM. As additional data is needed, the data in RAM is written to the disk and is replaced by new data from the data file on the disk. Consequently, the size of RAM has little effect on the size of the data file that can be processed.

Data Exchange

Although integrated products must be able to share or transfer data between their applications, the scope of the data exchange may be limited. One limitation concerns the flow of data between applications. For instance, the data transfer may be unidirectional, that is, data can be transferred from application A to application B but not vice versa. Or, the transfer may be bidirectional, that is, data can be transferred from application A to application B and then back to application A. A second limitation concerns the extent of the data exchange. Can each application within a product transfer data to another application? Some applications can only accept transferred data, but can not transfer their own data to other applications. For example, application A can accept transferred data from application B or C, but its data can not be transferred to either application B or C or any other application contained in the product.

Storage Allocation

Because the data formats and storage algorithms of integrated products differ, the size of a data file may vary. For efficiency, many vendors have developed data formats that are tailored to their products. These formats may incorporate special control characters and may involve unique coding representations, both may increase or decrease the disk space used. Additionally, vendors use different storage algorithms to save data files on disk. The storage algorithm affects the disk space occupied by a data file and the time to store or retrieve the file. The following spreadsheet example is used to illustrate two methods for saving a data file.

A spreadsheet contains 3 rows (A, B, C) and 4 columns $(1, 2, 3, 4)$.
- Row A, Column 1 contains the number 7 and

- Row C, Column 4 contains the number 11.
- All other cells (row and column positions) are empty.

	1	2	3	4
A	7			
B				
C				4

Some integrated products may save only those cells with information using a method called sparse matrices. Thus, two cells are saved, Row A, Column 1 ($A1$) and Row C, Column 4 ($C4$). Whereas other integrated products may save the entire spreadsheet, putting zero's (0) in the empty cells. Thus, 12 cells are saved, $A1, A2, A3, A4, B1, B2, B3, B4$, and $C1, C2, C3, C4$.

Product Bias

Most integrated products are designed with a bias towards one of its applications. The product either performs this application better than its others activities, is used by all others, or controls all others. The biases are:

- Application centered – either commands are structured around one application (other applications being a special case of the *main* application), or other applications rely on the *main* application for their data.
- Data centered – no application bias exists and one central data file is accessed by all the applications.
- Administrator centered – all applications must pass through and are controlled by a central program or manager.

Typically, a product's strength is commensurate with its orientation, that is, the application it does best will be the biased application. It is advisable to choose a product whose bias is consistent with the user's application requirements.

3.3.2 Product Features

Product features include technical attributes that reflect the capabilities of the product. The attributes include: application breadth, programmability, multitasking, windows, and hardware required or supported.

Application Breadth

The number and types of applications available or contained in the integrated products vary. Some products will allow the user to choose the applications to be integrated

and some products do not (Section 2.2). Users should insure that the applications currently available for a product will fulfill their application requirements.

Programmability

A programming capability provides a method for automating a sequence of instructions. The user is able to customize the integrated product to his needs by defining a series of commands that are executed with a minimum number of keystrokes. Integrated products can have either of the following types of programming.

- Keyboard macros – are the simplest kind of programming. User issued commands and keystrokes are recorded under a macro name and executed whenever that name is invoked. They are often used to automate repetitive tasks.

- Programming languages – are high level languages that contain both commands and logic control. For instance, these *programs* can branch to subroutines, perform conditional logic, or control error processing. They can be used to create new commands, customized templates, and user menus.

Multitasking

Multitasking is the ability to run two or more tasks simultaneously. Because of operating system limitations, only a few integrated products perform multitasking. Many of these products use a time slice approach, in which the computer processor cycles through the tasks, rapidly switching from one to another. Some integrated products can only multitask one task, printing. This allows the user to print one file while editing another. Caution: a product that loads multiple tasks into memory but freezes the background tasks, giving all processing time to the foreground task is not true multitasking.

Windows

Windows provide a means of dividing the screen into distinct regions, with each region capable of displaying information from disparate applications or from the same application. This ability to concurrently view information from multiple windows can enhance document (file) handling, e.g., transferring data from one file to another or looking at the beginning and end of a long file. Windows, however, offer some drawbacks if they are not well implemented or used effectively. Possible drawbacks include: slowed cursor movement, increased memory allocations, and limits on the amount of visible data. The number of windows, their placement on the screen (overlapping or side by side), movement, size, etc. vary among the integrated products.

Hardware Required or Supported

As with any software, integrated products are written to operate and support specific computers and peripherals. Moreover, the integrated products may require additional hardware such as a hard disk, a specified quantity of RAM, a color display,

and/or a graphics display. The type of printer and its capabilities must also be considered. An integrated product may not support a specific printer or may require a specific type of printer. If a printer is not fully supported, the full range of capabilities of the printer and integrated product may not be realized.

3.3.3 Quality

Quality includes the non-technical attributes of the product. The attributes include: learning aids, product quality, and vendor support.

Learning Aids

Learning aids reduce the time and effort a user spends on learning and using an integrated product efficiently and effectively. The learning aids can be provided by reference manuals, written and on-line tutorials, on-line help, error messages, command prompting, and user groups. User groups provide a forum for the exchange of information and often include product demonstrations, lectures, and expert panels to answer questions. Although some vendors furnish all these types of learning aids for their integrated products, others may provide a subset of these aids.

Product quality

The integrated product should be well engineered and free of software errors or *bugs*. It should be organized, clearly presented and perform efficiently. Examine the synergy of the product, in other words, how its parts (e.g., applications, command language, user interface) work together. Consider the intuitiveness of command names and the number of keystrokes required to invoke the command. It is often easier to remember a command in which the command name, the function it performs, and the keystroke used to invoke it have a common bond (e.g., start with the same letter). Finally, insure that the documentation for the product is complete and correct.

Vendor Support

It is not unusual for a software product to be announced and delivered late or to be announced and never produced. The latter is often called vaporware. The marketplace is very volatile and a product available today may not survive the competition and exist tomorrow. Before selecting a product, find out about the vendor's stability, reputation, product commitment, and responsiveness to customers. Will the vendor provide new product releases that fix software bugs, contain new features and enhancements, and support additional hardware? User support provided by the vendor is also important and may be in the form of telephone hot lines, newsletters, and local representatives.

4 CONCLUSIONS

Initially, it seemed as if integrated software for microcomputers would eventually replace standalone software products. New products were continually being announced, introduced into the marketplace and bought. Naturally, all products claimed to solve user computing needs and enable them to be more productive. Now however, the marketplace has calmed down; product limitations are being recognized, fewer new products are being introduced, and current products are being enhanced.

Many integrated products have been acquired with little or no regard for the user or the application requirements. As a result, owners of these products may be using a minimum number of the product's functions or have stopped using the product, reverting to previous methods of work. To remedy this situation, the benefits afforded and restrictions imposed by the integrated software must be examined to make certain that it can fit the user's working style and application requirements. Three key issues to consider are product limitations, user type, and software alternatives.

Limitations to an integrated product can be caused by the hardware configuration of the system as well as the product's design and capabilities. These limitations can include lack of support for a hardware component, degraded performance of a feature or capability, or size restrictions on the data files. Just because a product can perform a task, does not imply that it does it well or completely. Thus, it is important to examine what the product can and can not do.

Users can be classified as either *casual* users or *power* users. The casual user has need of several applications, jumps between the applications, and uses the *standard* set of features supplied by the software product. Whereas the power user will use a few applications, rarely switch or share data between the applications, and use the advanced features of the applications, including writing programs to customize the applications to his needs.. Integrated software may be more appropriate for the casual user than the power user.

Integrated products may not be the only software solution to user problems. Often user needs can be fulfilled by standalone software. Many of these products have excellent import/export features, that is, the ability to read data files created from other software products or write data files that can be read by other products. Moreover, standalone products may have the range of features and capabilities that best meet the application requirements now and in the future. It is important to consider both the integrated and standalone products as potential software solutions.

Although fewer new integrated products are being announced today, the marketplace is not stagnant. Current vendors are upgrading and improving the quality of their existing

products. These product changes range from minor bug fixes and support of new hardware to additional applications, improved interfaces with other software, and improved performance. The distinction between integrated products is becoming less clear as vendors are incorporating features and capabilitites into their products that were once unique to their competitor's product. Examples of this waning distinction include:

- adding applications and features, such as an outline processor, macro library, or word wrap,
- improving memory management by changing from memory oriented to disk oriented,
- changing the storage algorithm to a sparse matrix implementation.

Because of these revisions, many integrated products can now be placed into more than one integration approach category. For instance, some of the multifunction applications products have the ability to *add-in* applications, a characteristic of the product suite. And, as applications are written specifically for a software integrator product, that product would also be considered an operating environment. Thus, the distinction between the integration approaches remain the same, but the product examples are no longer attributed to a unique approach.

Over the next few years, products belonging to all five integration approaches will continue to be available. However, within each approach, a few favorite products will dominate, becoming defacto standards and dictating future directions. Look for products to gain power and versatility, acquire intelligence or the ability to correct and anticipate command sequences, and to be friendlier and more adaptive to a user's working styles.

5 REFERENCES

Barkley, John, Gilbert, D., and Hankinson, A., "Selection of Microcomputer Systems", *National Bureau of Standards, Special Publication* **500-112**, March 1984.

"End User's Guide to Buying Small Computers", *U.S. General Services Administration, Office of Information Resources Management*, July 1984.

Frankel, Sheila, "Introduction to Software Packages", *National Bureau of Standards Special Publication* **500-114**, April 1984.

Gilbert, Dennis, Parker, E., and Rosenthal, L., "Microcomputers: A Review of Federal Agency Experiences", *National Bureau of Standards, Special Publication* **500-102**, June 1983.

"Guidance on Requirements Analysis for Office Automation Systems", *National Bureau of Standards, Special Publication* **500-72**, December 1980.

Hecht, Myron, Hecht, H., and Press, L., "Microcomputers: Introduction to Features and Uses", *National Bureau of Standards, Special Publication* **500-110**, March 1984.

"Microcomputer Management Guidelines", *Federal Software Testing Center, Office of Software Development and Information Technology*, Report **OIT/FSTC-83/019**, November 1983.

Zawilski, Anthony, "Integrated Applications Software", *FOSE Software*, 1985.

NOTE: NBS Special Publications are available from the Government Printing Office, Washington, DC 20402.

APPENDIX

A PRODUCTS and VENDORS

This appendix provides a listing of several of the commercially available integrated products. It should be noted that this is by no means a comprehensive list of all integrated products. The products are categorized by integration approach as described in Section 2 of this report. It should also be noted that the absence or presence of a particular product does not imply endorsement or criticism by the National Bureau of Standards or the author.

MULTIFUNCTION APPLICATION

Product Names	Vendors
Ability	Xanaro Technologies
Appleworks	Haba Systems and Apple Computer Inc.
Aura	BPI Systems
Enable	The Software Group
Framework	Ashton-Tate
Goldengate	Cullinet
Integrated 7	Mosaic Software
Knowledgeman	MDBS
MBA, Corporate MBA	Context
1-2-3, Symphony, Jazz	Lotus

PRODUCT SUITES

Product Names	Vendors
InteSoft Series	Schuchardt Software Systems
IT Software	Martin Marrietta Data Systems
Personal Decision Series	IBM
Powerbase	Powerbase Systems
Prevail	Inspiration Systems Inc.
Smart Series	Innovative Software

SOFTWARE INTEGRATOR

Product Names	Vendors
Concurrent PC-DOS	Digital Research
DESQ	Quarterdeck Systems
Windows	Microsoft
Window Master	Structured Systems Group
3270 PC	IBM

OPERATING ENVIRONMENTS

Product Names	Vendors
GEM	Digital Research
Macintosh	Apple Computer Inc.
Starburst	Micropro
Topview	IBM
Windows	Microsoft

BACKGROUND UTILITIES

Product Names	Vendors
Desk Organizer	Warner Software
Get Organized	Electronic Arts
Macintosh Desk Assessories	Apple Computer Inc.
My Desk	Third Floor Systems
Pop-Ups	Bellsoft, Inc.
R Desk	R Systems Inc.
Sidekick	Borlund International
Spotlight	Lotus

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET** *(See instructions)*	1. PUBLICATION OR REPORT NO. NBS/SP-500/135	2. Performing Organ. Report No.	3. Publication Date January 1986

4. TITLE AND SUBTITLE

Computer Science and Technology:
Integrated Software for Microcomputer Systems

5. AUTHOR(S)

Lynne S. Rosenthal

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

National Bureau of Standards
Department of Commerce
Gaithersburg, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Same as item 6.

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 86-600500

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

Integrated software products combine several applications within a single program and enable information to be shared between the applications. This report defines five approaches to integration: the all-in-one, product suite, software integrator, operating environment, and background utility. Each of these approaches is designed to achieve different objectives by emphasizing the power and improtance of the features of each approach. Consequently, there is no best approach to software integration. The selection of an approach depends on the application requirements, current system configurations, and personal preferences. Selecting an integrated product begins by considering the various approaches to integration and determining which one is most appropriate. Subsequently, the products within the chosen approach are evaluated againist a preestablished set of criteria relating to the product design, technical capabilities, and product quality. Careful selection of an integrated product will insure that the benefits to be gained from its use can be achieved.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

All-in-one package; background utility; integrated software; integration; product suite; operating environment; software integrator.

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

38

15. Price

USCOMM-DC 6043-P80

*U.S. Government Printing Office: 1986 0 - 491-070 (265)

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

NBS Technical Publications

Periodical

Journal of Research—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).
NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.
Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*
Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards
Gaithersburg, MD 20899